Rough Interface Reconstruction Using the Level Set Method

Yootai Kim*
The Ohio State University

Raghu Machiraju†
The Ohio State University

David Thompson‡
Mississippi State University

ABSTRACT

We present a new level set method for reconstructing interfaces from point aggregations. Although level-set-based methods are advantageous because they can handle complicated topologies and noisy data, most tend to smooth the inherent roughness of the original data. Our objective is to enhance the quality of a reconstructed surface by preserving certain roughness-related characteristics of the original dataset. Our formulation employs the total variation of the surface as a roughness measure. The algorithm consists of two steps: a roughness-capturing flow and a roughness-preserving flow. The roughness capturing step attempts to construct a surface for which the original roughness is captured – distance flow is well suited for roughness capturing. Surface reconstruction is enhanced by using a total variation preserving (TVP) scheme for the roughness-preserving flow. The shock filter formulation of Osher and Rudin is exploited to achieve this goal. In practice, we have found that better results are obtained by balancing the TVP term with a smoothing term based on curvature. The algorithm is applied to both fractal surface growth simulations and scanned data sets to demonstrate the efficacy of our approach.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Surfaces and object representation

Keywords: point sampled data, surface reconstruction, level set method, shock filter, total variation preserving, rough surface

1 INTRODUCTION

Interface reconstruction from point aggregations is a challenging problem. An interface for a point aggregation is defined as a surface that contacts the void exterior to the aggregation. Such aggregations often imply *rough interfaces*. These interfaces are characterized as clusters of points possessing fractal dimensions. The problem of interface reconstruction has no unique solution. Furthermore, the interface is often not a manifold and its topology may be far from simple. In particular, if an interface has holes or overhangs, it is difficult to parameterize.

These point sets may be generated from studies of physical phenomena. In the material sciences, numerous discrete surface simulation models have been developed to study surface growth phenomena [3]. For instance, consider the simulation result (in 2D) using the diffusion limited aggregation (DLA) cluster growth model shown in Fig. 1. In addition to fractal-like micro-structures, prominent dendritic structures are produced. Note that the blue-colored curve in Fig. 1 represents the interface between the volumetric point aggregation and the void into which it extends. This is why we use the phrase *interface reconstruction* rather than surface reconstruction. Interface reconstruction includes two distinct steps – capturing the natural interface followed by traditional surface reconstruction.

*e-mail: yootai@cse.ohio-state.edu
†e-mail: raghu@cse.ohio-state.edu
‡e-mail: dst@erc.msstate.edu

Most surface reconstruction methods assume that input point data are sampled from a surface. In other words, the interface is already given for the problem. It should be noted that the point aggregations described here are volumetric in nature and are created from an evolution or growth of an interface. Interior points exist below the interface and are part of the aggregations. Hence, we need a method to both capture the interface and reconstruct it for for DLA data sets.

A second example, shown in Fig. 2, is a scan of a three-dimensional ice accretion that can occur on an aircraft wings. Such scans are used to catalog and characterize ice accretions for evaluating aircraft performance degradation in icing conditions. Though the data must be filtered before any kind of interface reconstruction can be performed, it is almost impossible to remove all noise. Hence, the ideal interface reconstruction algorithm should be tolerant to the presence of noise. It should be noted that the ice accretion process can be thought of as being the result of a more complex DLA-like process.
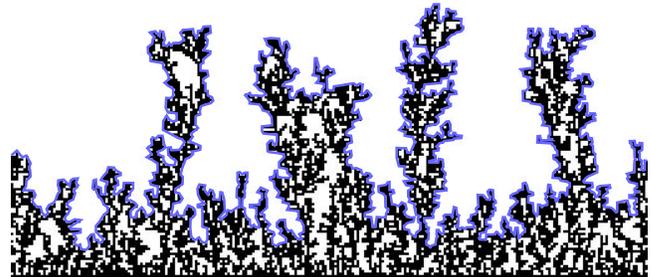

Figure 1: Point aggregation from a 2D DLA simulation. The blue line demarcates a possible interface.
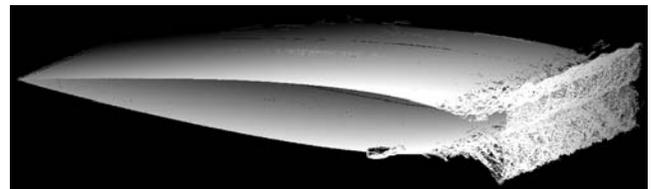

Figure 2: Point aggregation from an iced wing scan.

The two applications described above motivate the need for a robust methodology to capture and represent complex interfaces. Explicit methods of representation using triangulation are unable to capture the complexity of such an interface in its entirety. Additionally, their noise handling capabilities are limited. On the other hand, implicit methods do possess the capability to represent rough interfaces with complex topologies.

Level set methods, through appropriate choices of initial conditions and front velocities, can capture interfaces while preserv-

ing their underlying topologies. However, level-set-based reconstruction methods tend to smooth the inherent *roughness* existing in the underlying data. This shortcoming motivates our use of a roughness-preserving level set formulation that employs a total variation preserving filter. As we explain later, interface roughness can be characterized by the total variation of the surface.

Our roughness-preserving level set method consists of two steps: a roughness-capturing step and a roughness-preserving step. First, we find an initial surface using a distance field flow to capture the interface of the point set and minimize the effect of smoothing. Second, a level set filter comprised of a linear combination of a roughness-preserving term and a smoothing term is applied to the distance flow surface iteratively until specified conditions are satisfied. Novel to this approach is the use of an edge-enhancing shock filter in the roughness-preserving term. Important aspects of our approach include:

- The work reported here, and prior related work [9], represent the first applications of level-set-based algorithms for rough interface reconstruction from point aggregations.

- A shock filter is employed as a roughness-preserving term.

- A single parameter provides control of the overall effect of roughness and smoothness.

- A unified level set formulation combines interface capturing and roughness enhancement.

Our paper is organized as follows. In Section 2, we review related efforts in surface reconstruction. We then explain the roughness-preserving level set formulation in Section 3. The details of our implementation are provided in Section 4. In Section 5, we include results that demonstrate the effectiveness of our approach and in the final section we draw conclusions and present a discussion of future work.

2 RELATED WORK

There exist two classes of methods for reconstructing surfaces. There are those which employ explicit triangulation schemes and exploit the Voronoi diagram or its dual, the Delaunay triangulation [2,5]. These algorithms are interpolatory in nature. The *power crust* algorithm of [2] computes an approximate medial axis transform (MAT), then produces a piecewise-linear surface representation using an inverse transform of the MAT. The *power crust* algorithm is provably correct if the sampling density is high enough everywhere. However, the local topology can change after reconstruction and holes may appear from undersampling. The *cocone* algorithm [5] allows reconstruction from an input that is not sufficiently sampled by detecting the boundaries between densely sampled regions and undersampled regions. However, these methods are not suitable for point clouds with noise or volumetric point aggregations with interior point samples similar to the examples presented in the earlier section. Results obtained using these algorithms may not be closed or even well-defined surfaces.

The second class of methods are implicit in nature. Radial basis functions (RBF) [4] can be used for processing point clouds with local problems (holes and incomplete surfaces). The objective is to find a scalar function such that all data points that are close to an isocontour of that scalar function. Distance functions have also been used for reconstruction from point clouds. The implicit scalar function is the closest distance of grid points to surfaces of the data cloud. In [8], a signed distance function is first computed from an estimated tangent plane. Later, the zero isosurface of the level set function is extracted. However, to be applicable to our problem,

these methods would also require utilization of a sophisticated technique to identify the interface of the volumetric point aggregations.

The level set method was originally introduced by Osher and Sethian [14] for numerical interface evolution through curvature flow and has been successfully used to capture interfaces for wide variety of problems. See [12] for a comprehensive review. The level set method is a powerful iterative numerical technique for deforming implicit surfaces that works in any number of dimensions. The data structures employed are very simple, and topological changes in the underlying and evolving surface are handled easily in a natural way. Zhao et al. [23] proposed the weighted minimal surface model based on the distance potential functional for reconstructing surfaces from point clouds. An initial surface is continuously deformed toward a final surface along the gradient direction of the functional until a surface potential force and a surface tension force reach equilibrium. In [9], we use the level set method to extract interfaces from point data generated by DLA simulations. Our formulation used only an attraction term (based on the distance field) to deform the initial surface. Enright et al. [6] proposed a hybrid particle level set method to accurately capture the interface driven by a flow field. They exploit a marker-based particle Lagrangian scheme to overcome the inherent smoothing effect of a level set method.

Anisotropic diffusion methods are used to smooth surfaces while preserving or enhancing sharp geometric features such as edges or corners. These methods are closely related to level set methods. Preusser et al. [16] proposed an anisotropic geometric diffusion model for the level set method. They used a regularized shape operator depending on pre-smoothed principal curvatures and principal directions of curvature to identify important surface features so that the level set propagation speed is decreased near edges and corners. Tasdizen et al. used fourth-order level set flows instead of the common second-order processes [18–20]. They solve the anisotropic diffusion equation on the normal map of the surface, and deform the surface to fit the smoothed normals. While all these methods are intended for feature-preserving surface fairing, our method is specifically designed for extracting rough interfaces from volumetric point aggregations. In this context, our current effort is an improvement of the level set formulation presented in our previous work [9]. A shock filter is introduced with a curvature term to control the roughness of the reconstructed surface while the distance field term for the level set speed is used in [9]. The combination of a shock filter and diffusion term have been used in image restoration and deblurring applications [7,10].

The basic form of shock-diffusion formulation employed in those papers is:

$$I_t = -\alpha \operatorname{sign}(I_{xx})|I_x| + \beta I_{xx}, \qquad (1)$$

where $I \equiv I(x,t)$ is a scalar function in one dimension, and α and β are constant weight parameters. I_t, I_x and I_{xx} are time, first spatial and second spatial derivatives, respectively. This formulation is very similar to our TVP formulation. However, it is intended for image deblurring while ours is intended for interface reconstruction of point aggregations using a level set function. In addition, the dimension of the problem domain does not change in image deblurring problem. In contrast, our formulation is constructed in higher dimensions of the problem domain since we embed a surface into a level set function. Furthermore, our shock filter is motivated by the notion that surface roughness can be characterized by its total variation.

3 ROUGHNESS-PRESERVING LEVEL SET METHOD

Our roughness-preserving level set method employs two distinct level set formulations: a *roughness-capturing flow* and a *roughness-preserving flow*. Fig. 3 describes the pipeline of our roughness-
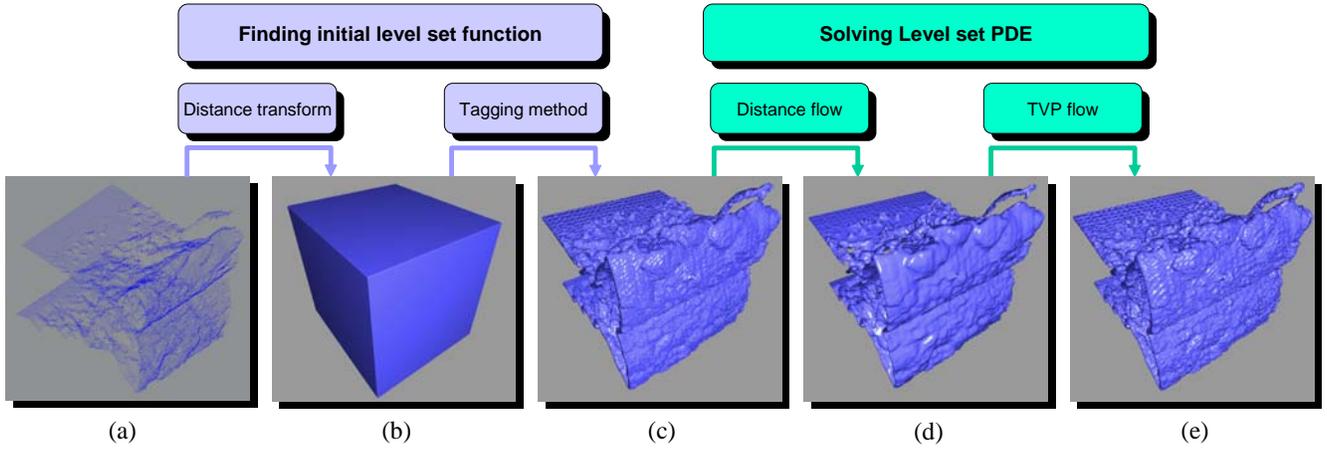
| (a) | (b) | (c) | (d) | (e) |

Figure 3: Pipeline of roughness-preserving level set reconstruction: (a) input point set, (b) distance function: the bounding surface is the initial surface for tagging method, (c) signed distance function: an approximate offset surface, (d) resulting surface of distance flow, (e) final surface

preserving level set reconstruction. We first apply the roughness-capturing flow to construct an initial surface that inherits the roughness of the actual interface with minimal smoothing. A distance flow formulation is used in the first stage since it allows the robust construction of a surface that closely approximates the true interface. The roughness-preserving flow is then applied to enhance the quality of the approximate surface by maintaining the roughness of the interface as it evolves. We employ total variation, which in this context is an integral of the magnitude of the first derivative of the level set function of the interface, as a measure of the roughness of the surface. This suggests that a total variation preserving(TVP) formulation should be employed for the roughness-preserving flow. In our approach, the TVP flow is realized using the shock filter of Osher and Rudin [13]. We now describe the shock filter and its inclusion in the level set formulation.

3.1 TVP and the Osher-Rudin Shock Filter

The total variation (TV) of a scalar function of one variable $u(x)$ can be defined as

$$TV(u) = \int |u_x| dx. \tag{2}$$

This definition can easily be extended to two or more spatial dimensions. TV can be employed as a measure of surface roughness since it can serve as a measure of the high frequency content of the surface. Fig. 4 shows the TV values for a simple, periodic two-dimensional function for a range of fundamental spatial frequencies. Clearly, the TV value increases as the *roughness* of the function grows.

The foundation of our method is the assertion that, by preserving the total variation of the surface as the level set method is applied to the distance flow surface, a more accurate representation of the surface will be obtained. Osher and Rudin proposed the use of a shock filter for edge enhancement in image processing [13]. They showed that their shock filter is total variation preserving in the discrete case.

The formulation for the shock filter is given by

$$u_t = -F(u_{xx})|u_x| \tag{3}$$

where $u \equiv u(x,t)$ is a scalar function in one dimension with the initial condition $u(x,0) = u_0(x)$. The Lipschitz continuous function
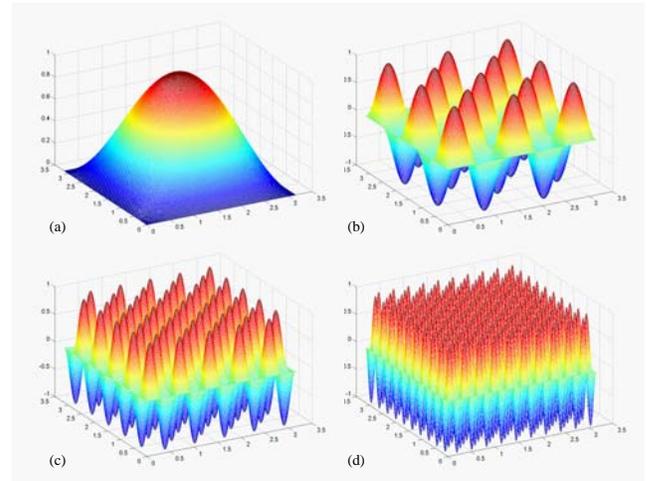

Figure 4: TV values for $u(x,y) = \sin(\omega x)\sin(\omega y)$: (a) $\omega = 1, TV = 6.6864$ (b) $\omega = 5, TV = 33.4319$, (c) $\omega = 10, TV = 66.8639$, (d) $\omega = 20, TV = 133.7277$

F should satisfy $F(0) = 0$ and $F(s)\text{sign}(s) \geq 0$. Choosing $F(s) = \text{sign}(s)$ gives the classical shock filter equation

$$u_t = -\text{sign}(u_{xx})|u_x| \tag{4}$$

The two-dimensional extension of the shock filter equation is given by

$$u_t = -F(L(u))|\nabla u| \tag{5}$$

where $u \equiv u(x,y,t)$ and $L(u)$ is a second-order nonlinear elliptic operator. $F(L(u))$ serves as an edge detection term. It changes sign across any singular feature so that the local flow is directed toward the features. For example, one popular choice for $L(u)$ is the Second-Derivative-in-the-Gradient-Direction (SDGD) operator $\nabla_{\eta\eta}u$ where η is the direction of the gradient. Hence, the common two-dimensional shock filter form is

$$u_t = -\text{sign}(\nabla_{\eta\eta}u)|\nabla u| \tag{6}$$

253

Key properties of the shock filter formulation include [7]:

- Shocks develop at inflection points.

- Local extrema remain unchanged in time.

- The scheme is TVP.

3.2 Level Set Formulation

The two salient steps of our level set formulation are described below:

Embed the surface A co-dimension one surface Γ is defined as the zero isosurface of a level set function $\phi(\mathbf{x})$, i.e., $\Gamma = \{\mathbf{x} : \phi(\mathbf{x}) = 0\}$. ϕ is negative inside Γ and positive outside Γ. In practice, the signed distance function is often the level set function of choice. Geometric properties of the surface Γ, such as the normal and mean curvature can be easily computed from ϕ using

$$\text{outward unit normal:} \quad \mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|} \quad (7)$$

$$\text{mean curvature:} \quad \kappa = \nabla \cdot \frac{\nabla\phi}{|\nabla\phi|} \quad (8)$$

Embed the motion The time evolution PDE for the level set function is obtained by differentiating $\phi(\Gamma(t), t)$ as

$$\phi_t + \frac{d\Gamma(t)}{dt} \cdot \nabla\phi = 0 \quad \Leftrightarrow \quad \phi_t + v_n|\nabla\phi| = 0 \quad (9)$$

Here, v_n is the normal directional speed of $\Gamma(t)$ which may depend on external physics or global and local geometric quantities. The PDE is solved for $\mathbf{x} \in \Omega$ ($\Omega \in \Re^3$ in our problem) and $t \in [0, T_f]$ with well-posed boundary conditions.

Define $d(\mathbf{x}) = \text{distance\_function}(\mathbf{x}, S)$ to be the closest distance between the position $\mathbf{x}$ and the input point data set S. The following distance flow is used for roughness capturing

$$\frac{d\phi}{dt} = d_s(\mathbf{x})|\nabla\phi| \quad (10)$$

where the signed distance function, $d_s(\mathbf{x})$, is:

$$d_s(\mathbf{x}) = \begin{cases} d(\mathbf{x}) & \text{if } \mathbf{x} \text{ is outside of the surface of } S \\ -d(\mathbf{x}) & \text{otherwise} \end{cases}$$

This distance-based speed term produces an attraction toward the true surface. In our previous work [9], only this term was used to reconstruct an interface from a DLA simulation point data set. We now describe the new roughness preserving formulation.

We can naturally extend Eq. 6 to the level set formulation for the roughness-preserving flow

$$\frac{d\phi}{dt} = -\text{sign}(\nabla_{\eta\eta}\phi)|\nabla\phi| \quad (11)$$

where η is the direction of the gradient. In practice, we found a better reconstruction can be achieved by balancing the TVP term with a surface smoothing term. Hence, we add a mean curvature speed term, often referred to as the surface tension or regularizing term, to the shock filter

$$\frac{d\phi}{dt} = -|\nabla\phi|\Big[\text{sign}(\nabla_{\eta\eta}\phi) - \kappa\Big] \quad (12)$$

In this formulation, the curvature term is likely dominate since it can be much larger than the TVP term whose magnitude is in the

interval $[0, 1]$. Therefore, we equalize the weights of the two front speed terms by normalizing κ using the following transformation

$$\tilde{\kappa} = \frac{\kappa}{1 + |\kappa|} \quad (13)$$

The normalization transforms the magnitude of the curvature values into the range $[0, 1]$ so that it proportionally contributes to the TVP term. Further, we provide users with control of the roughness by a simple weighting of the two speed terms using the parameter γ as shown below:

$$\frac{d\phi}{dt} = -|\nabla\phi|\Big[\gamma\,\text{sign}(\nabla_{\eta\eta}\phi) - (1-\gamma)\tilde{\kappa}\Big] \quad (14)$$

Note that the original TVP flow is recovered for $\gamma = 1$ and the pure mean curvature flow recovered for $\gamma = 0$. It should be noted that this formulation is not strictly TVP for $\gamma \neq 1$.

Finally, we provide a unified formulation using the time dependent Heaviside function(H) given by

$$\begin{aligned} \frac{d\phi}{dt} &= -|\nabla\phi|H(-t + T_d)\Big[\gamma\,\text{sign}(\nabla_{\eta\eta})\phi - (1-\gamma)\tilde{\kappa}\Big] \\ &+ |\nabla\phi|H(t - T_d)d(\mathbf{x}) \end{aligned} \quad (15)$$

where the Heaviside function(see Fig. 5) is defined by

$$H(s) = \begin{cases} 1 & \text{if } s \geq 0 \\ 0 & \text{if } s < 0 \end{cases}$$

For $t \leq T_d$, the distance flow term is used. The roughness preserv-
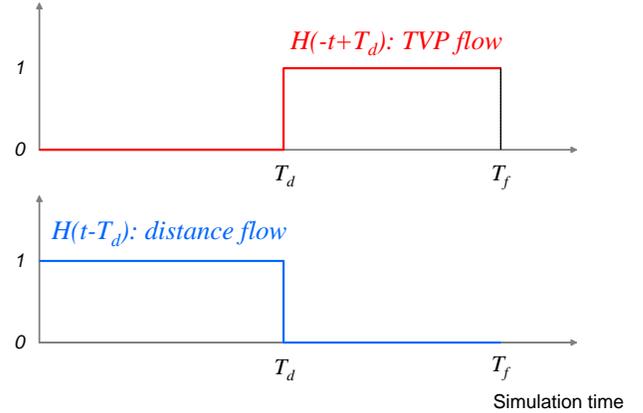

Figure 5: Heaviside functions

ing term is effective after the epoch T_d. T_d and T_f can be automatically determined using appropriate stopping criteria or are specified by the user.

4 IMPLEMENTATION

As shown in Fig. 3, we first obtain an initial surface by deforming the bounding surface following an approximate normal flow. This step is necessary because a good initial surface reduces the computational cost of solving the PDE. We then deform the approximate offset surface (Fig. 3(c)) to obtain the final surface by solving Eq. 15. A fast and stable numerical solver is critical to the success of this step. The final surface can be extracted as a polygonal model using the Marching Cubes method [11] and then rendered with standard graphics software.

4.1 Finding the Initial Surface

First, we compute the distance function to construct the level set function in which the initial surface is embedded. The distance function $d(\mathbf{x})$ to an input point data set S is computed by solving the following Eikonal equation

$$|\nabla d(\mathbf{x})| = 1, \qquad d(\mathbf{x}) = 0, \ \mathbf{x} \in S. \qquad (16)$$

We use the algorithm in [23] that combines upwind differencing with alternating direction Gauss Seidel iterations to solve the differential equation in Eq. 16.

The approximate offset surface is then used as an initial surface. To find such an offset surface, we use the simple tagging method described in [23]. We start from a surface that includes the true surface such as a bounding box(see Fig. 6(a)). Every grid cell is initially tagged as inside(i), boundary(b), or outside(o). Then, we deform the initial tagged boundary to the final offset surface. The deformation process is terminated when the maximum distance value of the tagged boundary is less than a specified offset distance. Fig. 6(b) shows a snapshot of the computational grid employed in the tagging process. After completing the tagging process, we can obtain the signed distance function by applying the distance computation algorithm to the tagged boundary.
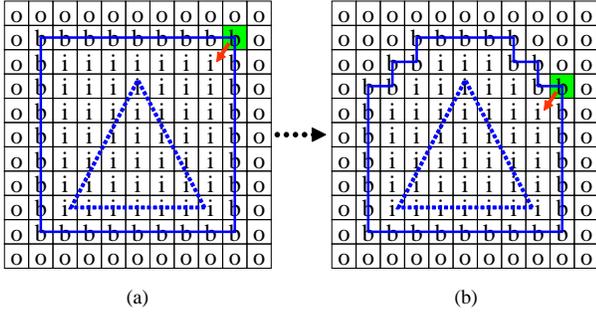

Figure 6: Tagging method: The solid line rectangle is an initial bounding box and the dotted line triangle is the true surface. A green cell is the one that has the maximum distance value. (a) initial status (b) updated status after a few iterations

4.2 Solving the Level Set PDE

We continuously deform the initial signed distance function, ϕ, by solving the level set PDE given by Eq. 9. If we solve the PDE for all computational grid nodes, the computational cost is $O(N^3)$ at each time step for a grid of size $N \times N \times N$. The computational cost reduces to $O(N^2)$ using the fast local level set method [15]. Instead of computing for every grid cell, the computation is performed only in a narrow tube around the zero level set (see Fig. 7). Since it is difficult to maintain the solution of Eq. 15 as a smooth signed distance function in the neighborhood of the front $\Gamma(t)$, a redistancing algorithm in [15] is employed.

First-order or higher-order ENO (Essentially Non-Oscillatory) type upwind schemes are used for spatial discretization of the advection term in Eq. 15. A central-difference approximation is employed for the SDGD edge filter term and the curvature term. Note that no boundary conditions are needed since they are enforced numerically by the upwind scheme. The first-order forward Euler scheme or higher-order TVD (Total Variation Diminishing) Runge-Kutta scheme is used for the temporal discretization. More details for the discretization scheme can be found in [14, 15, 22].
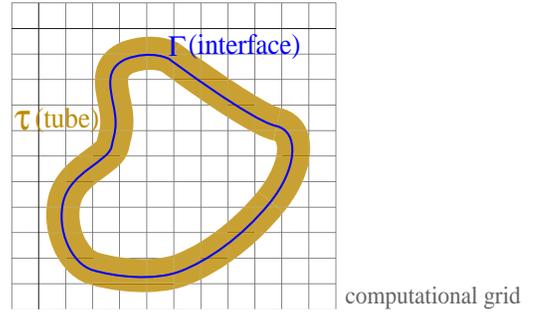

Figure 7: Local level set method: Computation is only performed on the narrow tube τ around the zero level set Γ

Since we concatenate two different level set PDEs, special attention is needed for the stability condition. We automatically determine the time step needed to ensure the stability condition for each level set PDE. The Courant-Friedreichs-Lewy condition (CFL condition) must be satisfied for the distance flow because it is a hyperbolic PDE. The CFL condition asserts that numerical waves should propagate no faster than physical waves. The parabolic stability condition is used for the TVP flow (Eq. 14) since the equation includes the second-order diffusion term (curvature term). Hence, the time step is determined by the following relations for the uniform grid

$$\text{distance flow (Eq. 10):} \quad \Delta t = \alpha_1 \frac{h}{\max_{p \in \tau}(v_p)}$$

$$\text{TVP flow (Eq. 14):} \qquad \Delta t = \alpha_2 h^2$$

where $\alpha_1, \alpha_2 \in (0,1)$, τ: the narrow tube around the frontal surface, v_p: the speed at the grid point p, h: grid spacing. Optimal α_1 and α_2 values vary depending on the properties of the data set. We use the following stopping criteria for the distance flow. We define the error measure e_i as the average of the distance between the zero level set for the iteration i and the data set. The simulation stops either if $e_i < tol$ or if $e_i > e_{i-1}$. In other words, the distance flow is terminated if either the surface is close enough to the data points or if the error begins to increase. The TVP flow executes for a given number of iterations without any stopping criteria until the desired enhancement is achieved.

5 RESULTS

The implementation of our level set solver exploits many of the utility classes available in the VISPACK library [21]. The zero level set, i.e., the desired implicit surface, is extracted using the Marching Cubes method as implemented in VTK [17]. The result is available as a polygon model format such as OBJ. Final images are rendered by MAYA [1]. The computations were conducted on a SGI Octane workstation with a MIPS R12000 processor and 500 MB of memory. We use the first-order solver to obtain the presented results. Results obtained using the second-order solver are visually similar to the first-order results and are not included here.

The reconstructions for the iced wing segment for a spectrum of roughness parameters are shown in Fig. 8. The computational grid is $198 \times 184 \times 176$. It takes approximately 28 seconds for each TVP iteration. Fig. 8(a) shows the original point set. Fig. 8(b) shows the initial surface obtained using the tagging method. Fig. 8(c) shows the result of applying the distance flow to the initial surface after 14 iterations. The remaining images demonstrate the effects of γ on

the roughness or smoothness of the final reconstructed interface. It is apparent that as value of γ becomes smaller, the resulting surface becomes smoother. The surface in Fig. 8(i) appears very smooth because pure curvature flow is used. In contrast, the result with only the TVP term shows a very rough surface in Fig. 8(d). Fig. 8(g) with $\gamma = 0.5$ shows a well-balanced result when compared to the other reconstructions.

Fig. 9 shows a rough surface generated by a three-dimensional DLA simulation. A $111 \times 101 \times 102$ computational grid is used. The computing time for each time step is approximately 4 sec. Fig. 9(b), 9(c) and 9(d) are the results for $\gamma = 1.0, 0.3$ and 0.0, respectively. Again, Fig. 9(b) shows more rugged surface than Fig. 9(d). Note that the roughness filtering sharpens the ridges of the initial surface as well as keeping the global shape.

6 CONCLUSIONS AND FUTURE WORK

We presented a new level-set based interface reconstruction technique for aggregate point clouds. We introduced a total variation preserving term as a roughness-preserving term in the level set formulation. Through a linear combination of the roughness-preserving term and the smoothing term, our method allows users to control the level of roughness of the final surface. Furthermore, we provide a single level set formulation that integrates an attraction term based on distance flow and the TVP/curvature flow. This integration allows for the creation of an intermediate surface from the application of a distance flow technique which is then provided for further enhancement by our TVP term. Noise (if any) is reduced through the inclusion of the curvature flow smoothing or regularizing term. We applied our technique to two representative data sets - a DLA simulation and a scanned wing with ice accretion.

Much future work remains. One imminent problem is that the TVP flow does not have a fitting term that keeps the surface close to the data points. The fitting term could be derived through a variational formulation based on a good roughness measure. Additionally, a rigorous analysis of our level set formulation is needed. In this paper, we employ the SDGD edge filter as a second-order elliptic operator in the term for the shock filter. It would be interesting to explore the effect of other edge operators such as a LoG (Laplacian-of-Gaussian) filter. In addition, feature-based values of the parameter γ may be employed to determine a better blend of the roughness preserving and smoothing terms. Finally, constraint based interface reconstruction is needed to efficiently handle open surfaces like the iced wings.

REFERENCES

[1] Alias Systems. *Maya 5 Unlimited*, 2003.

[2] N. Amenta, S. Choi, and R. K. Kolluri. The power crust, unions of balls, and the medial axis transform. *Computational Geometry*, 19(2-3):127–153, 2001.

[3] A. L. Barabási and H. E. Stanley. *Fractal Concepts In Surface Growth*. Cambridge University Press, Cambridge, 1995.

[4] J.C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans. Reconstruction and representation of 3d objects with radial basis functions. In *Proceedings of ACM SIG-GRAPH 2001*, Computer Graphics Proceedings, Annual Conference Series, pages 67–76. ACM, ACM Press / ACM SIGGRAPH, 2001.

[5] T. K. Dey and J. Giesen. Detecting undersampling in surface reconstruction. In *Proc. 17th Sympos. Comput. Geom.*, pages 257–263, 2001.

[6] D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell. A hybrid particle level set method for improved interface capturing. *J. Comput. Phys.*, 183:83–116, 2002.

[7] G. Gilboa, N. A. Sochen, and Y. Y. Zeevi. Regularized shock filters and complex diffusion. In *Proceedings of ECCV 2002 Vol. 1*, pages 399–413, 2002.

[8] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. In *Computer Graphics(Proceedings of ACM SIGGRAPH 92)*, pages 71–78. ACM, 1992.

[9] Y. Kim, R. Machiraju, and D. Thompson. Rough surface modeling using surface growth. In *Proceedings of 2003 International Conference on Shape Modeling and Applications*, pages 33–42, 2003.

[10] P. Kornprobst, R. Deriche, and G. Aubert. Image coupling, restoration and enhancement via pde's. In *Proceedings of International Conference on Image Processing 1997*, volume 2, pages 458–461, 1997.

[11] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *Computer Graphics(Proceedings of ACM SIGGRAPH 87)*, volume 21, pages 163–169. ACM, 1987.

[12] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer-Verlag, 2002.

[13] S. Osher and L. I. Rudin. Feature-oriented image enhancement using shock filters. *SIAM Journal of Numerical Analysis*, 27:919–940, 1990.

[14] S. Osher and J. Sethian. Fronts propagating with curvature dependent speed: Algorithms based in hamilton-jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.

[15] D. Peng, B. Merriman, H. Zhao, S. Osher, and M. Kang. A pde based fast local level set method. *Journal of Computational Physics*, 155:410–438, 1999.

[16] T. Preusser and M. Rumpf. A level set method for anisotropic geometric diffusion in 3d image processing. *SIAM Journal on Applied Mathematics*, 62(5):1772–1793, December 2002.

[17] W. Schroeder, K. Martin, and B. Lorensen. *The Visualization Toolkit An Object-Oriented Approach To 3D Graphics*. Kitware, Inc., third edition, 2003.

[18] T. Tasdizen and R. Whitaker. Anisotropic diffusion of surface normals for feature preserving surface reconstruction. In *Proceedings of Fourth International Conferenceon 3-D Digital Imaging and Modeling*, pages 353–360, 2003.

[19] T. Tasdizen and R. Whitaker. Feature preserving variational smoothing of terrain data. In *IEEE Workshop on Variational, Geometric and LevelSet Methods in Computer Vision*, 2003.

[20] T. Tasdizen, R. Whitaker, P. Burchard, and S. Osher. Geometric surface processing via normal maps. *ACM Trans. Graph.*, 22(4):1012–1033, 2003.

[21] R. T. Whitaker. Vispack:a c++ object oriented library for processing volumes, images, and level-set surface models. 2002.

[22] H. Zhao, T. Chan, B. Merriman, and S. Osher. A variational level set approach to multiphase motion. *Journal of Computational Physics*, 127:179–195, 1996.

[23] H. Zhao, S. Osher, and R. Fedkiw. Fast surface reconstruction using the level set method. In *Proceedings of IEEE Workshop on Variational and Level Set Methods in Computer Vision (VLSM 2001)*, pages 194–201, Jul 2001.
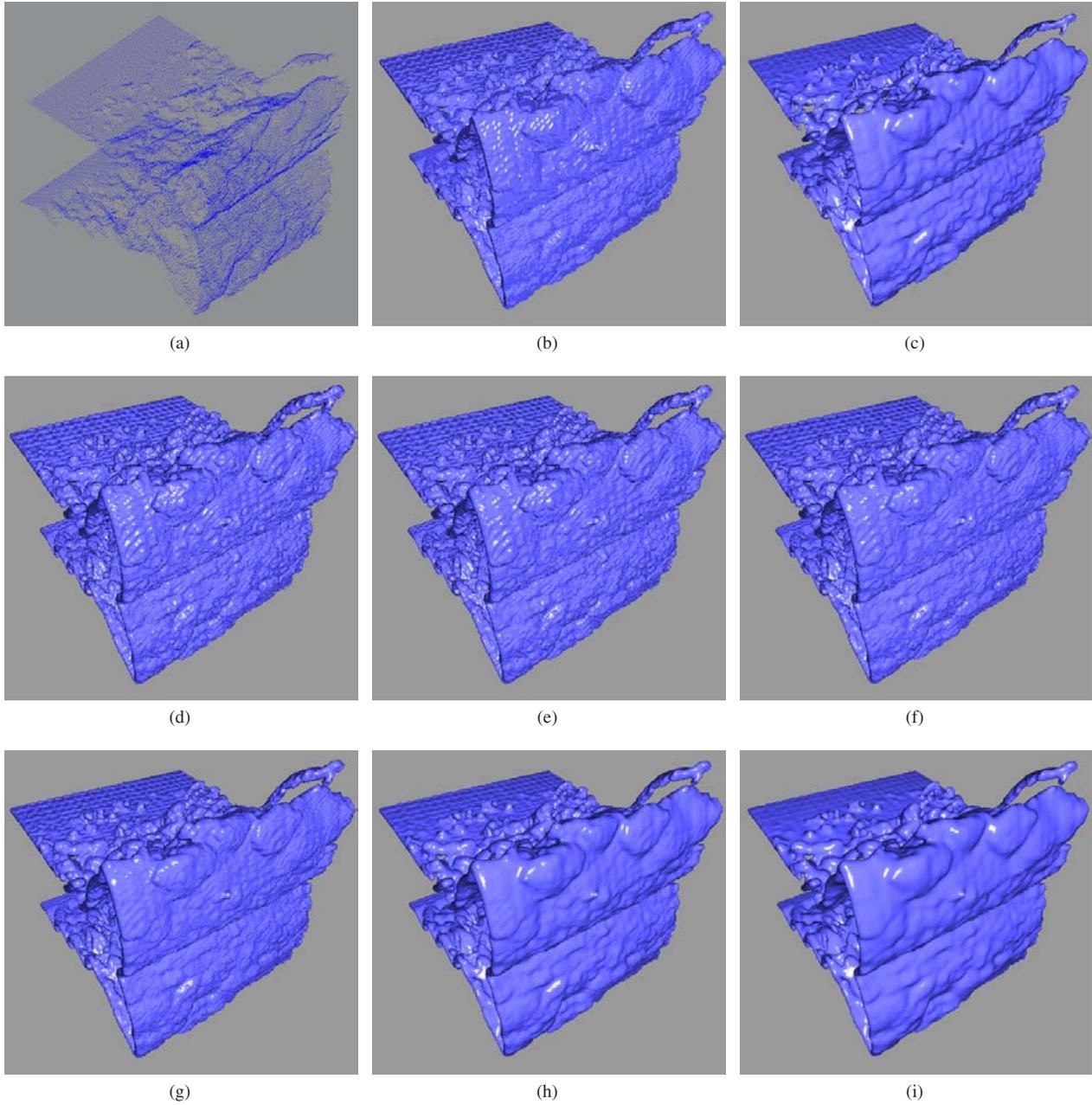
Figure 8: Reconstruction of the iced wing part: (a) point set, (b) initial surface, (c) distance flow with 14 iterations, (d) $\gamma = 1$ with 5 iterations, (e) $\gamma = 0.9$, (f) $\gamma = 0.7$, (g) $\gamma = 0.5$, (h) $\gamma = 0.3$, (i) $\gamma = 0.1$
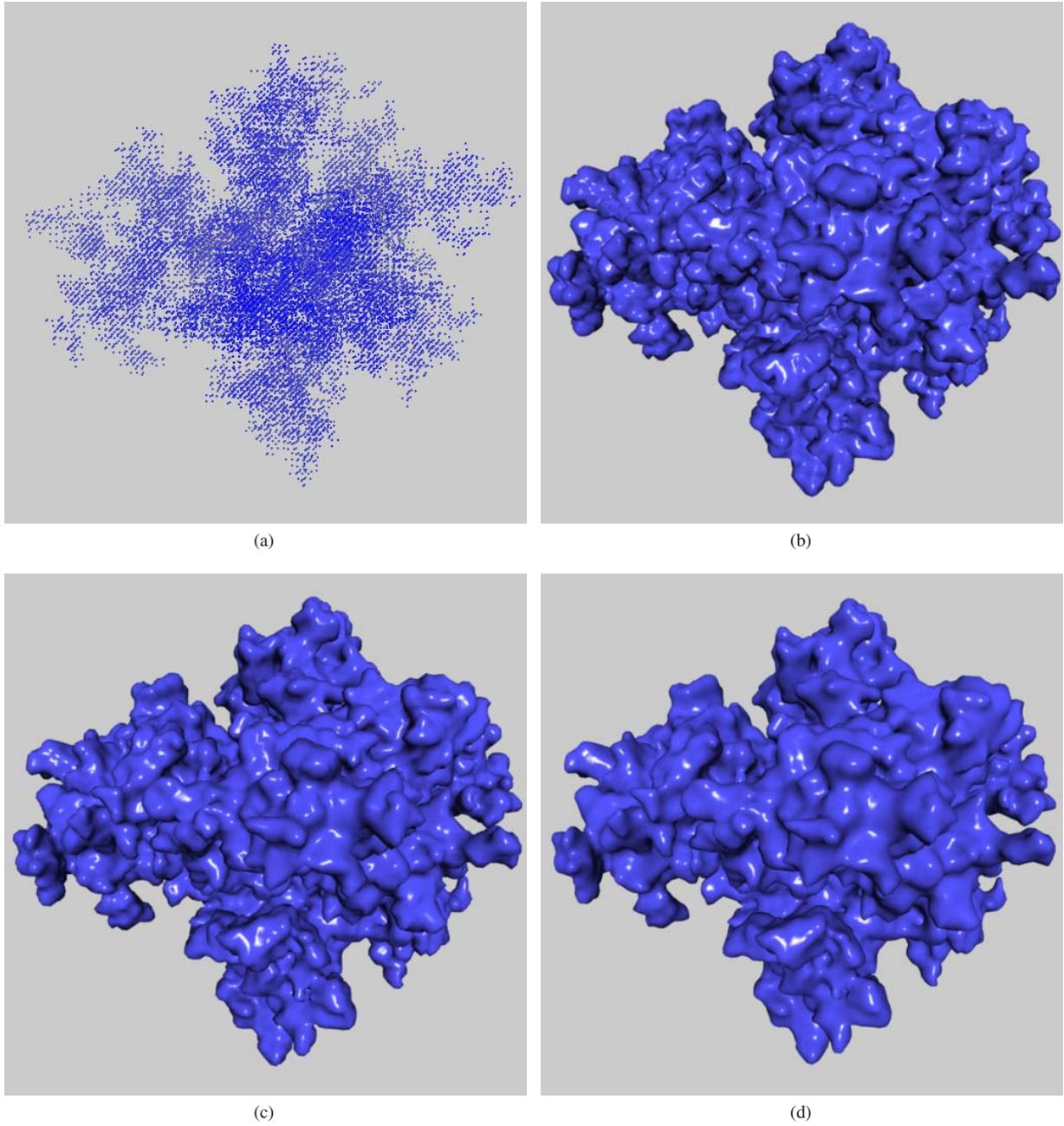
(a)

(b)

(c)

(d)

Figure 9: Reconstruction of 3D DLA point aggregation: (a) point set, (b) $\gamma = 1$ with 5 iterations, (c) $\gamma = 0.3$, (d) $\gamma = 0$